LISTING OF CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Cancelled)

2. (Previously Presented) The system of claim 17, wherein the speculative data load is loaded in the pipeline.

3. (Previously Presented) The system of claim 17, wherein one or more of the data loads in the pipeline are not dependent on any specific data load and not selectively flagged.

4. (Previously Presented) The system of claim 17, wherein the flag is a bit within the instruction.

5. (Previously Presented) The system of claim 17, wherein the flag is attached to the instruction.

6. (Previously Presented) The system of claim 17, wherein each flagged instruction is flushed from the pipeline upon the determination of a misprediction for a data load.

7. (Previously Presented) The system of claim 17, wherein the fast-load data cache includes a directory.

8. (Previously Presented) The system of claim 17, wherein the fast-load data cache does not include a directory.

9. (Cancelled)

10. (Previously Presented) The method of claim 19, further comprising the step of loading the speculative data load into the pipeline.

11. (Previously Presented) The method of claim 19, wherein the step of selectively flagging the one or more instructions does not flag an instruction that is not dependent on a specific instruction.

12. (Previously Presented) The method of claim 19, wherein the step of selectively flagging a dependant instruction occurs through altering a bit within the dependant instruction.

13. (Previously Presented) The method of claim 19, wherein the step of selectively flagging the dependant instruction occurs through attaching a flag to the dependant instruction.

14. (Previously Presented) The method of claim 19, further comprising the step of flushing the flagged dependent instruction from the pipeline upon the determination of a misprediction of a corresponding data load.

15-16. (Cancelled)

17. (Previously Presented) A computer architecture, comprising:

at least one pipeline able to selectively load, execute and flush a series of instructions and capable of selectively flagging each of the series of instructions with a flag to indicate dependence upon the load of a speculative instruction;

at least one fast-load data cache that loads at least one speculative data load relative to the speculative instruction into the pipeline;

at least one L1 data cache that loads at least one non-speculative data load, corresponding to the speculative instruction, into the pipeline a predetermined number of cycles after the fast-load data cache loads the speculative data load;

a circuit that determines if the speculative data load is a misprediction; and

a circuit that causes:

if the speculative load is a misprediction, the pipeline to inhibit execution of the speculative load and any instructions dependant thereon and to execute the non-speculative load and any instructions dependant thereon; and

if the speculative load is not a misprediction, the pipeline to execute the speculative load and any instructions dependant thereon and to inhibit the non-speculative load and any instructions dependant thereon.

18. (Cancelled)

19. (Previously Presented) A method for executing instructions in a computer architecture that includes a pipeline, the method comprising the actions of:

a. loading a speculative load into the pipeline;

b. loading a non-speculative load into the pipeline a predetermined number of cycles after the action of loading a speculative load;

c. if the speculative load was a misprediction, then invalidating the speculative load in the pipeline and executing the non-speculative load, otherwise executing the speculative load and invalidating the non-speculative load.